When Is Amplification Necessary for Composition in Randomized Query Complexity?

Shalev Ben-David
University of Waterloo, Canada
shalev.b@uwaterloo.ca

Mika Göös
Stanford University, CA, USA
goos@stanford.edu

Robin Kothari
Microsoft Quantum and Microsoft Research, Redmond, WA, USA
robin.kothari@microsoft.com

Thomas Watson
University of Memphis, TN, USA
Thomas.Watson@memphis.edu

Abstract

Suppose we have randomized decision trees for an outer function f and an inner function g. The natural approach for obtaining a randomized decision tree for the composed function $(f \circ g^n)(x^1, \ldots, x^n) = f(g(x^1), \ldots, g(x^n))$ involves amplifying the success probability of the decision tree for g, so that a union bound can be used to bound the error probability over all the coordinates. The amplification introduces a logarithmic factor cost overhead. We study the question: When is this log factor necessary? We show that when the outer function is parity or majority, the log factor can be necessary, even for models that are more powerful than plain randomized decision trees. Our results are related to, but qualitatively strengthen in various ways, known results about decision trees with noisy inputs.

2012 ACM Subject Classification Theory of computation → Oracles and decision trees

Keywords and phrases Amplification, composition, query complexity

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2020.28

Category RANDOM

Funding *Thomas Watson*: Supported by NSF grant CCF-1657377.

Acknowledgements We thank Badih Ghazi for interesting discussions about this work, and we thank anonymous reviewers for their comments.

1 Introduction

A deterministic decision tree for computing a partial function $f\colon \{0,1\}^n \to Z$ is a binary tree where each internal node is labeled with an index from $[n]$ and each leaf is labeled with an output value from Z. On input $x \in \{0,1\}^n$, the computation follows a root-to-leaf path where at a node labeled with index i, the value of x_i is queried and the path goes to the left child if $x_i = 0$ and to the right child if $x_i = 1$. The leaf reached on input x must be labeled with the value $f(x)$ (if the latter is defined). The cost of the decision tree is its depth, i.e., the maximum number of queries it makes over all inputs. The deterministic query complexity of f is the minimum cost of any deterministic decision tree that computes f. We will consider several more general models of decision trees (randomized, etc.), so we repurpose traditional complexity class notation to refer to the various associated query complexity measures. Since

P is the traditional complexity class corresponding to deterministic computation, we let $\mathsf{P}(f)$ denote the deterministic query complexity of f. (Some of the recent literature uses the notation $\mathsf{P}^{\mathsf{dt}}(f)$, but this paper deals exclusively with decision trees, so we drop the dt superscript.)

A randomized decision tree is a probability distribution over deterministic decision trees. Computing f with error ε means that for every input x (for which $f(x)$ is defined), the probability that the output is not $f(x)$ is at most ε. The cost of a randomized decision tree is the maximum depth of all the deterministic trees in its support. The randomized query complexity $\mathsf{BPP}_\varepsilon(f)$ is the minimum cost of any randomized decision tree that computes f with error ε. When we write $\mathsf{BPP}(f)$ with no ε specified, we mean $\varepsilon = 1/3$. A basic fact about randomized computation is that the success probability can be amplified, with a multiplicative overhead in cost, by running several independent trials and taking the majority vote of the outputs: $\mathsf{BPP}_\varepsilon(f) \leq O(\mathsf{BPP}(f) \cdot \log(1/\varepsilon))$. See [9] for a survey of classic results on query complexity.

If $f\colon \{0,1\}^n \to Z$ and $g\colon \{0,1\}^m \to \{0,1\}$ are two partial functions, their composition is $f \circ g^n\colon (\{0,1\}^m)^n \to Z$ where $(f \circ g^n)(x^1, \ldots, x^n) \coloneqq f(g(x^1), \ldots, g(x^n))$ (which is defined iff $g(x^i)$ is defined for all i and $f(g(x^1), \ldots, g(x^n))$ is defined). How does the randomized query complexity of $f \circ g^n$ depend on the randomized query complexities of f and g? A simple observation is that to design a randomized decision tree for $f \circ g^n$, we can take a 1/6-error randomized decision tree for f and replace each query – say to the i^{th} input bit of f – with a $1/6n$-error randomized decision tree for evaluating $g(x^i)$. By a union bound, with probability at least 5/6 all of the (at most n) evaluations of g return the correct answer, and so with probability at least 2/3 the final evaluation of f is also correct. Since $\mathsf{BPP}_{1/6n}(g) \leq O(\mathsf{BPP}_{1/n}(g))$, we can write this upper bound as

$$\mathsf{BPP}(f \circ g^n) \;\leq\; O(\mathsf{BPP}(f) \cdot \mathsf{BPP}_{1/n}(g)) \;\leq\; O(\mathsf{BPP}(f) \cdot \mathsf{BPP}(g) \cdot \log n). \tag{1}$$

When is this tight? It will take some effort to suitably formulate this question. We begin by reviewing known related results.

1.1 When is amplification necessary?

As for general lower bounds (that hold for all f and g), much work has gone into proving lower bounds on $\mathsf{BPP}(f \circ g^n)$ in terms of complexity measures of f and g that are defined using models more powerful than plain randomized query complexity [16, 3, 6, 4, 5]. In terms of just $\mathsf{BPP}(f)$ and $\mathsf{BPP}(g)$, the state-of-the-art is that $\mathsf{BPP}(f \circ g^n) \geq \Omega(\mathsf{BPP}(f) \cdot \sqrt{\mathsf{BPP}(g)})$ for all f and g [14]. Furthermore, it is known that the latter bound is sometimes tight: There exist partial boolean functions f and g such that $\mathsf{BPP}(f \circ g^n) \leq \widetilde{O}(\mathsf{BPP}(f) \cdot \sqrt{\mathsf{BPP}(g)})$ and $\mathsf{BPP}(f), \mathsf{BPP}(g) \geq \omega(1)$ [14, 5]. Thus (1) is far from being *always* tight, even without worrying about the need for amplification. However, it remains plausible that $\mathsf{BPP}(f \circ g^n) \geq \Omega(\mathsf{BPP}(f) \cdot \mathsf{BPP}(g))$ holds for all *total* f and all partial g. We take this as a working conjecture in this paper. This conjecture has been confirmed for some specific outer functions f, such as the identity function $\textsc{Id}\colon \{0,1\}^n \to \{0,1\}^n$ [21] (this is called a "direct sum" result) and the boolean functions $\textsc{Or}$, $\textsc{Xor}$ (parity), and $\textsc{Maj}$ (majority) [17]. These results, however, do not address the need for amplification in the upper bound (1). To formulate our question of whether (1) is tight, a first draft could be:

Question A, with respect to a particular f: Is (1) tight for *all* partial functions g?

This is not quite a fair question, for at least two reasons:

- Regarding the first inequality in (1): The simple upper bound actually shows $\mathsf{BPP}(f \circ g^n) \leq O(\mathsf{BPP}(f) \cdot \mathsf{BPP}_{1/\mathsf{BPP}(f)}(g))$ (the union bound is only over queries that take place, not over all possible queries). So for simplicity, let us restrict our attention to f satisfying $\mathsf{BPP}(f) \geq \Omega(n)$, which is the case for Id, Or, Xor, and Maj.
- Regarding the second inequality in (1): Some functions g satisfy $\mathsf{BPP}_{1/n}(g) \leq o(\mathsf{BPP}(g) \cdot \log n)$ (e.g., if $\mathsf{P}(g) \leq O(\mathsf{BPP}(g))$). So for simplicity, let us restrict our attention to g satisfying $\mathsf{BPP}_{1/n}(g) \geq \Omega(\mathsf{BPP}(g) \cdot \log n)$, which (as we show later) is the case for two partial functions GapOr and GapMaj defined as follows ($|x|$ denotes the Hamming weight of $x \in \{0,1\}^m$):

$$\textsc{GapOr}(x) \coloneqq \begin{cases} 0 & \text{if } |x| = 0 \\ 1 & \text{if } |x| = m/2 \end{cases} \quad \text{and} \quad \textsc{GapMaj}(x) \coloneqq \begin{cases} 0 & \text{if } |x| = m/3 \\ 1 & \text{if } |x| = 2m/3 \end{cases}.$$

Thus, a better formulation of Question A would be: Assuming $\mathsf{BPP}(f) \geq \Omega(n)$, is (1) tight for all partial g satisfying $\mathsf{BPP}_{1/n}(g) \geq \Omega(\mathsf{BPP}(g) \cdot \log n)$? Even with these caveats, the answer is always "no." It will be instructive to examine a counterexample. Let $\textsc{Which}\colon \{0,1\}^2 \to \{0,1\}$ be the partial function such that $\textsc{Which}(y)$ indicates the location of the unique 1 in y, under the promise that $|y| = 1$. Then $g = \textsc{Which} \circ \textsc{GapOr}^2$ takes an input of length $2m$ with the promise that there are exactly $m/2$ many 1s, either all in the left half or all in the right half, and outputs which half has the 1s. It turns out $\mathsf{BPP}(g) \leq O(1)$ and $\mathsf{BPP}_{1/n}(g) \geq \Omega(\log n)$ provided $m \geq \log n$ (for similar reasons as GapOr itself) and yet $\mathsf{BPP}(f \circ g^n) \leq O(\mathsf{BPP}(f))$ for all f: To compute $f \circ g^n$, we can run an optimal randomized decision tree for f and whenever it queries $g(x^i)$, we repeatedly query uniformly random bit positions of x^i until we find a 1 (so the value of $g(x^i)$ is determined by which half we found a 1 in). This has the same error probability as the randomized decision tree for f, and the total number of queries to the bits of $(x^1, \ldots, x^n)$ is $O(\mathsf{BPP}(f))$ in expectation, because for each i it takes $O(1)$ queries in expectation to locate a 1 in x^i. By Markov's inequality, with high constant probability this halts after only $O(\mathsf{BPP}(f))$ total queries. Thus by aborting the computation if it attempts to make too many queries, we obtain a randomized decision tree for $f \circ g^n$ that always makes $O(\mathsf{BPP}(f))$ queries, with only a small hit in the error probability.

Blais and Brody [7] adjust the statement of Question A so the answer becomes "yes" in the case $f = \textsc{Id}$. Specifically, they weaken the right-hand side in such a way that the above counterexample is ruled out. Defining[1] $\overline{\mathsf{BPP}}_\varepsilon(g)$ similarly to $\mathsf{BPP}_\varepsilon(g)$ but where the cost of a randomized decision tree is the maximum over all inputs (on which g is defined) of the expected number of queries, we now have $\overline{\mathsf{BPP}}_{1/n}(g) \leq \overline{\mathsf{BPP}}_0(g) \leq O(1)$ for the g from the counterexample. The theorem from [7] is $\mathsf{BPP}(f \circ g^n) \geq \Omega(\mathsf{BPP}(f) \cdot \overline{\mathsf{BPP}}_{1/n}(g))$ when $f = \textsc{Id}$, in other words, $\mathsf{BPP}(g^n) = \Omega(n \cdot \overline{\mathsf{BPP}}_{1/n}(g))$ (a "strong direct sum" result). [7] also explicitly asked whether similar results hold for other functions f. The corresponding conjecture for $f = \textsc{Or}$ is false (as we note below) while for $f = \textsc{Xor}$ and $f = \textsc{Maj}$ it remains open.

To make progress, we step back and ask a seemingly more innocuous version of the question:

Question B, with respect to a particular f: Is (1) tight for *some* partial function g?

It turns out the answer is "no" for $f = \textsc{Or}$ and is "yes" for both $f = \textsc{Xor}$ and $f = \textsc{Maj}$.

[1] [7] used the notation $\overline{\mathsf{R}}$ instead of $\overline{\mathsf{BPP}}$.

1.2 Decision trees with noisy inputs

Question B is related to "query complexity with noisy inputs" (introduced in [13]), so let us review the latter model: When input bit y_i is queried, the wrong bit value is returned to the decision tree with some probability $\leq 1/3$ (and the correct value of y_i is returned with the remaining probability). The "noise events" are independent across all queries, including multiple queries to the same input bit. Now the adversary gets to pick not only the input, but also the "noise probabilities." [13] distinguishes between two extreme possibilites: A static adversary has a single common noise probability for all queries, while a dynamic adversary can choose a different noise probability for each node in the decision tree. In this paper we make a reasonable compromise: The adversary gets to choose a tuple of noise probabilities $(\nu_1, \dots, \nu_n)$, and each query to y_i returns $1 - y_i$ with probability exactly ν_i. When a randomized decision tree computes f with error probability ε, that means for every input $y \in \{0,1\}^n$ and every noise probability tuple $(\nu_1, \dots, \nu_n)$ (with $\nu_i \leq 1/3$ for each i), the output is $f(y)$ with probability $\geq 1 - \varepsilon$ over the random noise and randomness of the decision tree. We invent the notation $\mathsf{BPP}^*(f)$ for the minimum cost of any randomized decision tree that computes f on noisy inputs, with error probability $1/3$. We have $\mathsf{BPP}^*(f) \leq O(\mathsf{BPP}(f) \cdot \log n) \leq O(n \log n)$ by repeating each query $O(\log n)$ times and taking the majority vote (to drive the noise probabilities down to $o(1/n)$), and using a union bound to absorb the noise probabilities into the error probability. The connection with composition is that $\mathsf{BPP}(f \circ g^n) \leq \mathsf{BPP}^*(f) \cdot \mathsf{BPP}(g)$, because to design a randomized decision tree for $f \circ g^n$, we can take a $1/3$-error randomized decision tree for f with noisy inputs, and replace each query – say to y_i – with a $1/3$-error randomized decision tree for evaluating $g(x^i)$.

There is a similar connection for 1-sided error and 1-sided noise. When a randomized decision tree has 1-sided error ε, that means on 0-inputs the output is wrong with probability 0, and on 1-inputs the output is wrong with probability at most ε. We let $\mathsf{RP}(g)$ denote the minimum cost of any randomized decision tree that computes g with 1-sided error $1/2$. Similarly, 1-sided noise means that when input bit y_i is queried, if the actual value is $y_i = 0$ then 1 is returned with probability 0, and if the actual value is $y_i = 1$ then 0 is returned with probability $\nu_i \leq 1/2$. We invent the notation $\mathsf{BPP}^\dagger(f)$ for the minimum cost of any randomized decision tree that computes f on 1-sided noisy inputs, with 2-sided error probability $1/3$. We have $\mathsf{BPP}(f) \leq \mathsf{BPP}^\dagger(f) \leq \mathsf{BPP}^*(f)$. The connection $\mathsf{BPP}(f \circ g^n) \leq \mathsf{BPP}^\dagger(f) \cdot \mathsf{RP}(g)$ holds like in the 2-sided noise setting. We officially record these observations:

▶ **Observation 1.** *For all f and g,*

$$\mathsf{BPP}(f \circ g^n) \;\leq\; \mathsf{BPP}^*(f) \cdot \mathsf{BPP}(g) \quad \textit{and} \quad \mathsf{BPP}(f \circ g^n) \;\leq\; \mathsf{BPP}^\dagger(f) \cdot \mathsf{RP}(g).$$

The upshot is that noisy upper bounds imply composition upper bounds, and composition lower bounds imply noisy lower bounds. There are many proofs of the result $\mathsf{BPP}^*(\textsc{Or}) \leq O(n)$ [13, 23, 25, 20]:

▶ **Theorem 2** ($\textsc{Or}$ never necessitates amplification)**.** $\mathsf{BPP}^*(O_R) \leq O(n)$ *and thus for every partial function g,*

$$\mathsf{BPP}(O_R \circ g^n) \;\leq\; O(n \cdot \mathsf{BPP}(g)).$$

Theorem 2 is not new, but in Appendix A we provide a particularly clean and elementary proof (related to, but more streamlined than, the proof in [23]). We mention that the proof straightforwardly generalizes to some other functions f, such as "odd-max-bit": $\textsc{Omb}(y) = 1$ iff the highest index of any 1 in y is odd.

We turn our attention to lower bounds. Various special-purpose techniques have been developed for proving query complexity lower bounds in the noisy setting [13, 12, 11, 20]. However, a conceptual consequence of Observation 1 is that special-purpose techniques are not generally necessary: We can just use techniques for lower bounding plain (non-noisy) randomized query complexity, applied to composed functions.

1.3 Lower bound for parity

[13] proved that $\mathsf{BPP}^*(\textsc{Xor})$ and $\mathsf{BPP}^*(\textsc{Maj})$ are $\Omega(n \log n)$. Although apparently not recorded in the literature, it is possible to generalize this result to show $\mathsf{BPP}^\dagger(\textsc{Xor})$ and $\mathsf{BPP}^\dagger(\textsc{Maj})$ are $\Omega(n \log n)$. However, we prove results even stronger than that, using the composition paradigm. Our results involve query complexity models that are more powerful than BPP, and even more powerful than the $\overline{\mathsf{BPP}}$ model from [7]. This follows a theme from a lot of prior work: Since BPP query complexity is rather subtle, we can make progress by studying related models that are somewhat more "well-behaved."

- As observed in [7], the $\overline{\mathsf{BPP}}$ model is equivalent to one where the cost is the worst-case (rather than expected) number of queries, and a randomized decision tree is allowed to abort (i.e., output a special symbol $\perp$) with at most a small constant probability, and the output should be correct with high probability conditioned on not aborting.
- If we strengthen the above model by allowing the non-abort probability to be arbitrarily close to 0 (rather than close to 1), but require that the non-abort probabilities are approximately the same for all inputs (within some factor close to 1), the resulting model has been called $\mathsf{2WAPP}$ ("2-sided weak almost-wide PP") [18, 17]. The "1-sided" version WAPP, defined later, will be relevant to us.
- If we further strengthen the model by allowing the non-abort probabilities to be completely unrelated for different inputs (and still arbitrarily close to 0), the resulting model has been called $\mathsf{PostBPP}$ ("BPP with post-selection") [18, 10].

We first consider the last of these models. $\mathsf{PostBPP}_\varepsilon(f)$ is the minimum cost of any randomized decision tree such that on every input x (for which $f(x)$ is defined), the probability of outputting $\perp$ is < 1, and the probability of outputting $f(x)$ is $\geq 1 - \varepsilon$ conditioned on not outputting $\perp$. Trivially, $\mathsf{PostBPP}(f) \leq \mathsf{BPP}(f)$. In fact, the $\mathsf{PostBPP}$ model is much more powerful than plain randomized query complexity; for example (noted in [18]) it can efficiently compute the aforementioned odd-max-bit function: $\mathsf{PostBPP}(\textsc{Omb}) \leq 1$.

For the noisy input setting, $\mathsf{PostBPP}^*$ and $\mathsf{PostBPP}^\dagger$ are defined in the natural way, and $\mathsf{PostBPP}(f \circ g^n) \leq \mathsf{PostBPP}^*(f) \cdot \mathsf{BPP}(g)$ and $\mathsf{PostBPP}(f \circ g^n) \leq \mathsf{PostBPP}^\dagger(f) \cdot \mathsf{RP}(g)$ hold like in Observation 1.

In Section 2 we prove something qualitatively much stronger than $\mathsf{BPP}^*(\textsc{Xor}) \geq \Omega(n \log n)$:

▶ **Theorem 3** ($\textsc{Xor}$ sometimes necessitates amplification). *For some partial function g, namely $g = \textsc{GapMaj}$ with $m \geq \log n$,*

$$\mathsf{PostBPP}(\textsc{Xor} \circ g^n) \;\geq\; \Omega(n \cdot \mathsf{BPP}_{1/n}(g)) \;\geq\; \Omega(n \log n \cdot \mathsf{BPP}(g)).$$

In particular, $\mathsf{PostBPP}^*(\textsc{Xor}) \geq \Omega(n \log n)$.

Let us compare Theorem 3 to two previous results.

- [12] proved that $\overline{\mathsf{BPP}}^*(\textsc{Xor}) \geq \Omega(n \log n)$ and that this lower bound holds even in the average-case setting (i.e., $\Omega(n \log n)$ queries are needed in expectation to succeed with high probability over a uniformly random input, random noise, and randomness of the decision tree). Our proof of Theorem 3 is simpler than the proof in [12] (though both proofs have a Fourier flavor), it also works in the average-case setting, and it yields a stronger result since the model is $\mathsf{PostBPP}$ instead of just $\overline{\mathsf{BPP}}$ (and the lower bound holds for composition rather than just noisy inputs). [11] presented a different simplified proof of the result from [12], but that proof does not generalize to $\mathsf{PostBPP}^*$.
- Our proof of Theorem 3 shows something analogous, but incomparable, to the strong direct sum from [7]. As we explain in Section 2, our proof shows that $\mathsf{PostBPP}(\textsc{Xor} \circ g^n) \geq \Omega(n \cdot \mathsf{PostBPP}_{1/n}(g))$ holds for *all* g (thus addressing a version of our Question A). Compared to the [7] result that $\overline{\mathsf{BPP}}(\textsc{Id} \circ g^n) \geq \Omega(n \cdot \overline{\mathsf{BPP}}_{1/n}(g))$ for all g, our result has the advantages of working for $f = \textsc{Xor}$ rather than $f = \textsc{Id}$ and yielding a qualitatively stronger lower bound ($\mathsf{PostBPP}$ rather than $\overline{\mathsf{BPP}}$ on the left side), but the disadvantage of also requiring the qualitatively stronger type of lower bound on g. Our result shows that if amplifying g requires a log factor in a very strong sense (even $\mathsf{PostBPP}$-type decision trees cannot avoid the log factor), then that log factor will be necessary when composing $\textsc{Xor}$ with g.

1.4 Lower bound for majority

Our main result strengthens the bound $\mathsf{BPP}^*(\textsc{Maj}) \geq \Omega(n \log n)$ from [13], mainly by holding for the stronger model WAPP (rather than just BPP), but also by directly handling 1-sided noise and by holding for composition rather than just noisy inputs.

$\mathsf{WAPP}_\varepsilon(f)$ is the minimum cost of any randomized decision tree such that for some $t > 0$, on input x the probability of outputting 1 is in the range $[(1-\varepsilon)t, t]$ if $f(x) = 1$, and in the range $[0, \varepsilon t]$ if $f(x) = 0$. The ε subscript should always be specified, because unlike BPP and $\mathsf{PostBPP}$, WAPP is not amenable to efficient amplification of the error parameter ε [18]. For every constant $0 < \varepsilon < 1/2$, we have $\mathsf{PostBPP}(f) \leq O(\mathsf{WAPP}_\varepsilon(f)) \leq O(\mathsf{BPP}(f))$.

WAPP-type query complexity has several aliases, such as "approximate conical junta degree" and "approximate query complexity in expectation," and it has recently played a central role in various randomized query (and communication) complexity lower bounds [22, 18, 16, 17]. One can think of WAPP as a nonnegative version of approximate polynomial degree (which corresponds to the class AWPP); in other words, it is a classical analogue of the polynomial method used to lower bound quantum algorithms.

For the noisy input setting, WAPP^* and $\mathsf{WAPP}^\dagger$ are defined in the natural way, and $\mathsf{WAPP}_\varepsilon(f \circ g^n) \leq \mathsf{WAPP}^*_\varepsilon(f) \cdot \mathsf{BPP}(g)$ and $\mathsf{WAPP}_\varepsilon(f \circ g^n) \leq \mathsf{WAPP}^\dagger_\varepsilon(f) \cdot \mathsf{RP}(g)$ hold like in Observation 1. We prove the following theorem, which shows that WAPP sometimes requires amplification, even in the one-sided noise setting.

▶ **Theorem 4** ($\textsc{Maj}$ sometimes necessitates amplification)**.** *For some partial function g, namely $g = \textsc{GapOr}$ with $m \geq \log n$, and some constant $\varepsilon > 0$,*

$$\mathsf{WAPP}_\varepsilon(\textit{Maj} \circ g^n) \;\geq\; \Omega(n \cdot \mathsf{BPP}_{1/n}(g)) \;\geq\; \Omega(n \log n \cdot \mathsf{RP}(g)).$$

In particular, $\mathsf{WAPP}^\dagger_\varepsilon(\textit{Maj}) \geq \Omega(n \log n)$.

This theorem should be contrasted with the work of Sherstov about making polynomials robust to noise [27]. In that work, Sherstov showed that approximate polynomial degree never requires a log factor in the noisy input setting, nor in composition. That is to say, he

improved the simple bound $\mathsf{AWPP}^*(f) \le O(\mathsf{AWPP}(f) \cdot \log n)$ to $\mathsf{AWPP}^*(f) \le O(\mathsf{AWPP}(f))$ for all Boolean functions f, and showed $\mathsf{AWPP}(f \circ g^n) \le O(\mathsf{AWPP}(f) \cdot \mathsf{AWPP}(g))$. In contrast, for conical juntas (nonnegative linear combinations of conjunctions), Theorem 4 shows that in a strong sense, the simple bound $\mathsf{WAPP}^*_\epsilon(f) \le O(\mathsf{WAPP}_\delta(f) \cdot \log n)$ (for all constants $0 < \delta < \varepsilon < 1/2$ and total Boolean functions f) cannot be improved: $\mathsf{WAPP}^\dagger_\varepsilon(f) \ge \Omega(\mathsf{WAPP}_0(f) \cdot \log n)$ for some constant ε and some total f, namely $f = \textsc{Maj}$. Thus unlike polynomials, conical juntas cannot be made robust to noise.

Our proof of Theorem 4 (in Section 3) introduces some technical ideas that may be useful for other randomized query complexity lower bounds.

By a simple reduction, Theorem 4 for $g = \textsc{GapOr}$ implies the same for $g = \textsc{GapMaj}$ (with $\mathsf{BPP}(g) = 1$ instead of $\mathsf{RP}(g) = 1$ at the end of the statement), but we do not know of a simpler direct proof for the latter result. Theorem 4 cannot be strengthened to have $\mathsf{PostBPP}$ in place of WAPP, because $\mathsf{PostBPP}(\textsc{Maj} \circ \textsc{GapMaj}^n) \le O(n)$. However, Theorem 4 does hold with $\textsc{Xor}$ in place of $\textsc{Maj}$, by the same proof.

2 Proof of Theorem 3: Xor sometimes necessitates amplification

We first discuss a standard technique for proving randomized query complexity lower bounds, which will be useful in the proof of Theorem 3. For any conjunction $C\colon \{0,1\}^k \to \{0,1\}$ and distribution $\mathcal{D}$ over $\{0,1\}^k$, we write $C(\mathcal{D}) \coloneqq \mathbb{E}_{x\sim\mathcal{D}}[C(x)] = \mathbb{P}_{x\sim\mathcal{D}}[C(x) = 1]$. The number of literals in a conjunction is called its width.

▶ **Fact 5.** *Let $h\colon \{0,1\}^k \to \{0,1\}$ be a partial function, and for each $z \in \{0,1\}$ let $\mathcal{D}_z$ be a distribution over $h^{-1}(z)$. Then for every ε there exist a conjunction C of width $\mathsf{PostBPP}_\varepsilon(h)$ and a $z \in \{0,1\}$ such that $\varepsilon \cdot C(\mathcal{D}_z) \ge (1-\varepsilon) \cdot C(\mathcal{D}_{1-z})$ and $C(\mathcal{D}_z) > 0$.*

Proof. Abbreviate $\mathsf{PostBPP}_\varepsilon(h)$ as r. Fix a randomized decision tree of cost r computing h with error ε conditioned on not aborting, and assume w.l.o.g. that for each outcome of the randomness, the corresponding deterministic tree is a perfect tree with 2^r leaves, all at depth r. Consider the probability space where we sample input x from the mixture $\frac{1}{2}\mathcal{D}_0 + \frac{1}{2}\mathcal{D}_1$, sample a deterministic decision tree T as an outcome of the randomized decision tree, and sample a uniformly random leaf ℓ of T. Let A be the indicator random variable for the event that ℓ is the leaf reached by $T(x)$ and its label is $h(x)$. Let B be the indicator random variable for the event that ℓ is the leaf reached by $T(x)$ and its label is $1 - h(x)$. Conditioned on any particular x and T, the probability that ℓ is the leaf reached by $T(x)$ is 2^{-r}. Thus conditioned on any particular x, if the non-abort probability is $t_x > 0$ then $\mathbb{E}[A \,|\, x] \ge 2^{-r} t_x (1-\varepsilon)$ and $\mathbb{E}[B \,|\, x] \le 2^{-r} t_x \varepsilon$ and thus $\varepsilon \cdot \mathbb{E}[A \,|\, x] - (1-\varepsilon) \cdot \mathbb{E}[B \,|\, x] \ge 0$. Over the whole probability space, we have $\varepsilon \cdot \mathbb{E}[A] - (1-\varepsilon) \cdot \mathbb{E}[B] \ge 0$, so by linearity the same must hold conditioned on some particular T and ℓ with $\mathbb{E}[A \,|\, T, \ell] > 0$. Let C be the conjunction of width r such that $C(x) = 1$ iff $T(x)$ reaches ℓ, and let z be the label of ℓ. Then we have $C(\mathcal{D}_z) = \mathbb{E}[A \,|\, T, \ell \text{ and } h(x) = z] = 2 \cdot \mathbb{E}[A \,|\, T, \ell] > 0$ and similarly $C(\mathcal{D}_{1-z}) = 2 \cdot \mathbb{E}[B \,|\, T, \ell]$. Thus

$$\varepsilon \cdot C(\mathcal{D}_z) - (1-\varepsilon) \cdot C(\mathcal{D}_{1-z}) \;=\; 2 \cdot \big(\varepsilon \cdot \mathbb{E}[A \,|\, T, \ell] + (1-\varepsilon) \cdot \mathbb{E}[B \,|\, T, \ell]\big) \;\ge\; 0. \qquad \blacktriangleleft$$

Now we work toward proving Theorem 3. Throughout, n is the input length of $\textsc{Xor}$, and m is the input length of $\textsc{GapMaj}$. We have $\mathsf{BPP}(\textsc{GapMaj}) \le 1$ by outputting the bit at a uniformly random position from the input. We describe one way of seeing that $\mathsf{BPP}_{1/n}(\textsc{GapMaj}) \ge \mathsf{PostBPP}_{1/n}(\textsc{GapMaj}) \ge \Omega(\log n)$ provided $m \ge \log n$. For $z \in \{0,1\}$, define $\mathcal{G}_z$ as the uniform distribution over $\textsc{GapMaj}^{-1}(z)$.

▶ **Fact 6.** *For every conjunction* $C\colon \{0,1\}^m \to \{0,1\}$ *of width* $w \le m/7$ *and for each* $z \in \{0,1\}$*,*

$$C(\mathcal{G}_z) \;\le\; 3^w \cdot C(\mathcal{G}_{1-z}).$$

Proof. By symmetry we just consider $z = 0$. Suppose C has u positive literals and v negative literals ($u + v = w$). Then

$$C(\mathcal{G}_0) \;=\; \tbinom{m-w}{m/3-u}/\tbinom{m}{m/3} \;\le\; \tbinom{m-w}{m/3}/\tbinom{m}{m/3} \;=\; \tfrac{(2m/3)\cdot(2m/3-1)\cdots(2m/3-w+1)}{m\cdot(m-1)\cdots(m-w+1)} \;\le\; (2/3)^w,$$

$$\begin{aligned} C(\mathcal{G}_1) \;=\; \tbinom{m-w}{m/3-v}/\tbinom{m}{m/3} \;\ge\; \tbinom{m-w}{m/3-w}/\tbinom{m}{m/3} \;&=\; \tfrac{(m/3)\cdot(m/3-1)\cdots(m/3-w+1)}{m\cdot(m-1)\cdots(m-w+1)} \\ &\ge\; \left(\tfrac{m/3-w}{m-w}\right)^w \;\ge\; \left(\tfrac{m/3-m/7}{m-m/7}\right)^w \;=\; (2/9)^w. \end{aligned}$$

Thus $C(\mathcal{G}_0)/C(\mathcal{G}_1) \le \left(\frac{2/3}{2/9}\right)^w = 3^w$. ◀

Combining Fact 5 and Fact 6 (using $h = \textsc{GapMaj}$, $k = m$, $\mathcal{D}_z = \mathcal{G}_z$, $\varepsilon = 1/n$, and $w = \mathsf{PostBPP}_\varepsilon(h)$) implies that $(1-\varepsilon)/\varepsilon \le 3^w$, in other words we have $\mathsf{PostBPP}_{1/n}(\textsc{GapMaj}) \ge \log_3(n(1-1/n)) \ge \Omega(\log n)$, provided $w \le m/7$. If $w > m/7$ then $\mathsf{PostBPP}_{1/n}(\textsc{GapMaj}) \ge \Omega(\log n)$ holds anyway provided $m \ge \log n$.

Hence, our result can be restated as follows.

▶ **Theorem 3 (Restated).** $\mathsf{PostBPP}(\textsc{Xor} \circ \textsc{GapMaj}^n) \ge \Omega(n \log n)$ *provided* $m \ge \log n$.

Proof. We show $\mathsf{PostBPP}(\textsc{Xor} \circ \textsc{GapMaj}^n) > \frac{1}{14} n \log n$. By Fact 5 (using $h = \textsc{Xor} \circ \textsc{GapMaj}^n$, $k = nm$, and $\varepsilon = 1/3$) it suffices to exhibit for each $z \in \{0,1\}$ a distribution $\mathcal{D}_z$ over $(\textsc{Xor} \circ \textsc{GapMaj}^n)^{-1}(z)$, such that for every conjunction C of width $\le \frac{1}{14} n \log n$ and for each $z \in \{0,1\}$, either $C(\mathcal{D}_z) < 2C(\mathcal{D}_{1-z})$ or $C(\mathcal{D}_z) = 0$. Letting $\mathcal{F}_z$ be the uniform distribution over $\textsc{Xor}^{-1}(z)$, define $\mathcal{D}_z$ as the mixture over $y \sim \mathcal{F}_z$ of $\mathcal{G}_y \coloneqq \mathcal{G}_{y_1} \times \cdots \times \mathcal{G}_{y_n}$ (i.e., $(x^1, \ldots, x^n) \sim \mathcal{G}_y$ is sampled by independently sampling $x^i \sim \mathcal{G}_{y_i}$ for all i). Put succinctly, $\mathcal{D}_z \coloneqq \mathbb{E}_{y \sim \mathcal{F}_z}[\mathcal{G}_y]$. Letting $\mathcal{G} \coloneqq \frac{1}{2}\mathcal{G}_0 + \frac{1}{2}\mathcal{G}_1$ and $\mathcal{F} \coloneqq \frac{1}{2}\mathcal{F}_0 + \frac{1}{2}\mathcal{F}_1$ and $\mathcal{D} \coloneqq \frac{1}{2}\mathcal{D}_0 + \frac{1}{2}\mathcal{D}_1$, we have $\mathcal{D} = \mathcal{G}^n$ since $\mathcal{F}$ is uniform over $\{0,1\}^n$. Since $C(\mathcal{D}) = \frac{1}{2}C(\mathcal{D}_0) + \frac{1}{2}C(\mathcal{D}_1)$, our goal of showing "$\frac{1}{2}C(\mathcal{D}_0) < C(\mathcal{D}_1) < 2C(\mathcal{D}_0)$ or $C(\mathcal{D}_0) = C(\mathcal{D}_1) = 0$" is equivalent to showing "$\frac{2}{3}C(\mathcal{D}) < C(\mathcal{D}_1) < \frac{4}{3}C(\mathcal{D})$ or $C(\mathcal{D}) = 0$."

Now consider any conjunction C of width $w \le \frac{1}{14} n \log n$ such that $C(\mathcal{D}) > 0$, and write $C(x^1, \ldots, x^n) = \prod_i C_i(x^i)$ where C_i is a conjunction. Since $C_i(\mathcal{G}) = \frac{1}{2}C_i(\mathcal{G}_0) + \frac{1}{2}C_i(\mathcal{G}_1)$, for each $y_i \in \{0,1\}$ we can write $C_i(\mathcal{G}_{y_i}) = (1 + a_i(-1)^{y_i})C_i(\mathcal{G})$ for some number a_i with $|a_i| \le 1$ (so $a_i \ge 0$ iff $C_i(\mathcal{G}_0) \ge C_i(\mathcal{G}_1)$). Let w_i be the width of C_i, so $\sum_i w_i = w \le \frac{1}{14} n \log n$. Then $w_i \le \frac{1}{7}\log n \le m/7$ for at least $n/2$ many values of i, and for such i note that by Fact 6, $C_i(\mathcal{G}_{y_i}) \le 3^{(\log n)/7} \cdot C_i(\mathcal{G}_{1-y_i}) \le n^{1/4} \cdot C_i(\mathcal{G}_{1-y_i})$ for each $y_i \in \{0,1\}$. The latter implies that $|a_i| \le 1 - 2/(n^{1/4}+1) \le 1 - n^{-1/4}$. Thus

$$\left|\textstyle\prod_i a_i\right| \;=\; \textstyle\prod_i |a_i| \;\le\; (1 - n^{-1/4})^{n/2} \;\le\; e^{-n^{3/4}/2} \;\le\; 1/4.$$

For $S \subseteq [n]$, let $\chi_S\colon \{0,1\}^n \to \{1,-1\}$ be the character $\chi_S(y) \coloneqq \prod_{i \in S}(-1)^{y_i} = (-1)^{\sum_{i \in S} y_i}$. Note that $\mathbb{E}_{y \sim \mathcal{F}_1}[\chi_S]$ is 1 if $S = \emptyset$, is -1 if $S = [n]$, and is 0 otherwise. Putting everything together,

$$\begin{aligned} C(\mathcal{D}_1) \;&=\; \mathbb{E}_{y \sim \mathcal{F}_1}[C(\mathcal{G}_y)] \;=\; \mathbb{E}_{y \sim \mathcal{F}_1}\left[\textstyle\prod_i C_i(\mathcal{G}_{y_i})\right] \;=\; \mathbb{E}_{y \sim \mathcal{F}_1}\left[\textstyle\prod_i (1 + a_i(-1)^{y_i})C_i(\mathcal{G})\right] \\ &=\; \left(\textstyle\prod_i C_i(\mathcal{G})\right) \cdot \mathbb{E}_{y \sim \mathcal{F}_1}\left[\textstyle\sum_{S \subseteq [n]} \prod_{i \in S} a_i(-1)^{y_i}\right] \\ &=\; C(\mathcal{D}) \cdot \textstyle\sum_{S \subseteq [n]} \left(\prod_{i \in S} a_i\right) \cdot \mathbb{E}_{y \sim \mathcal{F}_1}[\chi_S(y)] \\ &=\; C(\mathcal{D}) \cdot \left(1 - \textstyle\prod_{i \in [n]} a_i\right) \;\in\; C(\mathcal{D}) \cdot (1 \pm 1/4) \end{aligned}$$

which implies $\frac{2}{3}C(\mathcal{D}) < C(\mathcal{D}_1) < \frac{4}{3}C(\mathcal{D})$ since we are assuming $C(\mathcal{D}) > 0$. This concludes the proof of Theorem 3. ◀

Using strong LP duality (as in [15]), it can be seen that Fact 5 is a tight lower bound method up to constant factors: $\mathsf{PostBPP}_\varepsilon(h) \geq \Omega(c)$ iff it is possible to prove this via Fact 5 by exhibiting "hard input distributions" $\mathcal{D}_0$ and $\mathcal{D}_1$ (as we did for GapMaj in Fact 6). Since this was the only property of g used in the proof of Theorem 3, this implies that $\mathsf{BPP}(\textsc{Xor}\circ g^n) \geq \mathsf{PostBPP}(\textsc{Xor}\circ g^n) \geq \Omega(n\cdot\mathsf{PostBPP}_{1/n}(g))$ holds for all g, as we mentioned in Subsection 1.3.

3 Proof of Theorem 4: Maj sometimes necessitates amplification

We first discuss a standard technique for proving randomized query complexity lower bounds, which will be useful in the proof of Theorem 4. For any conjunction $C\colon \{0,1\}^k \to \{0,1\}$ and distribution $\mathcal{D}$ over $\{0,1\}^k$, we write $C(\mathcal{D}) \coloneqq \mathbb{E}_{x\sim\mathcal{D}}[C(x)] = \mathbb{P}_{x\sim\mathcal{D}}[C(x)=1]$. The number of literals in a conjunction is called its width.

▶ **Fact 7.** *Let $h\colon \{0,1\}^k \to \{0,1\}$ be a partial function, and let $\mathcal{D}_0$, $\mathcal{D}_1$, $\mathcal{D}_2$ be three distributions, over $h^{-1}(0)$, $h^{-1}(1)$, and $h^{-1}(0)\cup h^{-1}(1)$ respectively. Then for every $0 < \varepsilon \leq 1/10$ there exists a conjunction C of width $\mathsf{WAPP}_\varepsilon(h)$ such that $C(\mathcal{D}_0) \leq \delta\cdot C(\mathcal{D}_1)$ and $C(\mathcal{D}_2) \leq (1+\delta)\cdot C(\mathcal{D}_1)$ and $C(\mathcal{D}_1) > 0$, where $\delta \coloneqq 2\sqrt{\varepsilon}$.*

The key calculation underlying the proof of Fact 7 is encapsulated in the following:

▶ **Fact 8.** *Let P_0, P_1, P_2 be three jointly distributed nonnegative random variables with $\mathbb{E}[P_1] > 0$. For any $0 < \varepsilon \leq 1/10$, if $\mathbb{E}[P_0] \leq \varepsilon$ and $\mathbb{E}[P_1] \geq 1-\varepsilon$ and $\mathbb{E}[P_2] \leq 1$, then there exists an outcome o such that $P_0(o) \leq \delta\cdot P_1(o)$ and $P_2(o) \leq (1+\delta)\cdot P_1(o)$ and $P_1(o) > 0$, where $\delta \coloneqq 2\sqrt{\varepsilon}$.*

Proof of Fact 8. Let $W \coloneqq \{o : P_1(o) > 0\} \neq \emptyset$. Suppose for contradiction that for every outcome $o \in W$, either $P_0(o) > \delta\cdot P_1(o)$ or $P_2(o) > (1+\delta)\cdot P_1(o)$. Then W can be partitioned into events U and V such that $P_0(o) > \delta\cdot P_1(o)$ for every $o \in U$ and $P_2(o) > (1+\delta)\cdot P_1(o)$ for every $o \in V$. Letting I_U and I_V be the indicator random variables for these events, we have $\mathbb{E}[P_1\cdot I_U] + \mathbb{E}[P_1\cdot I_V] = \mathbb{E}[P_1]$ and thus either:

- $\mathbb{E}[P_1\cdot I_U] \geq \sqrt{\varepsilon}\cdot\mathbb{E}[P_1]$, in which case

$$\mathbb{E}[P_0] \;\geq\; \mathbb{E}[P_0\cdot I_U] \;>\; \delta\cdot\mathbb{E}[P_1\cdot I_U] \;\geq\; \delta\cdot\sqrt{\varepsilon}\cdot(1-\varepsilon) \;=\; 2\varepsilon(1-\varepsilon) \;>\; \varepsilon, \quad \text{or}$$

- $\mathbb{E}[P_1\cdot I_V] \geq (1-\sqrt{\varepsilon})\cdot\mathbb{E}[P_1]$, in which case

$$\mathbb{E}[P_2] \;\geq\; \mathbb{E}[P_2\cdot I_V] \;>\; (1+\delta)\cdot\mathbb{E}[P_1\cdot I_V] \;\geq\; (1+\delta)\cdot(1-\sqrt{\varepsilon})\cdot(1-\varepsilon) \;>\; 1$$

where the last inequality can be verified by a little calculus for $0 < \varepsilon \leq 1/10$.
Both cases yield a contradiction. ◀

Proof of Fact 7. Abbreviate $\mathsf{WAPP}_\varepsilon(h)$ as r. Fix a randomized decision tree of cost r computing h with error parameter ε and threshold $t > 0$ (from the definition of WAPP), and assume w.l.o.g. that for each outcome of the randomness, the corresponding deterministic tree is a perfect tree with 2^r leaves, all at depth r. Consider the probability space where we sample a deterministic decision tree T as an outcome of the randomized decision tree, and sample a uniformly random leaf ℓ of T. For any outcome T, ℓ, let $C_{T,\ell}$ be the conjunction of width r such that $C_{T,\ell}(x) = 1$ iff $T(x)$ reaches ℓ. Define three joint random variables P_0, P_1, P_2 as

$$P_j(T,\ell) \;\coloneqq\; \begin{cases} C_{T,\ell}(\mathcal{D}_j) & \text{if the label of } \ell \text{ is } 1 \\ 0 & \text{if the label of } \ell \text{ is } 0 \end{cases}.$$

Conditioned on any particular x and T, the probability that ℓ is the leaf reached by $T(x)$ is 2^{-r}. Thus

$$\begin{aligned}\mathbb{E}[P_j] &= \mathbb{P}_{T,\ell,\,x\sim\mathcal{D}_j}[\ell \text{ is the leaf reached by } T(x) \text{ and its label is } 1] \\ &= \mathbb{E}_{x\sim\mathcal{D}_j}\left[2^{-r}\cdot\mathbb{P}_T[T(x) \text{ outputs } 1]\right]\end{aligned}$$

which implies $\mathbb{E}[P_0] \le 2^{-r}t\varepsilon$ and $\mathbb{E}[P_1] \ge 2^{-r}t(1-\varepsilon)$ and $\mathbb{E}[P_2] \le 2^{-r}t$. Applying Fact 8 to the scaled random variables $(2^r/t)P_0$, $(2^r/t)P_1$, $(2^r/t)P_2$ yields an outcome T, ℓ such that

$$P_0(T,\ell) \le \delta\cdot P_1(T,\ell) \quad\text{and}\quad P_2(T,\ell) \le (1+\delta)\cdot P_1(T,\ell) \quad\text{and}\quad P_1(T,\ell) > 0.$$

Since $P_1(T,\ell) > 0$, the label of ℓ must be 1, so we get

$$C_{T,\ell}(\mathcal{D}_0) \le \delta\cdot C_{T,\ell}(\mathcal{D}_1) \quad\text{and}\quad C_{T,\ell}(\mathcal{D}_2) \le (1+\delta)\cdot C_{T,\ell}(\mathcal{D}_1) \quad\text{and}\quad C_{T,\ell}(\mathcal{D}_1) > 0.$$

◀

Now we work toward proving Theorem 4. Throughout, n is the input length of Maj, and m is the input length of GapOr. We have $\mathsf{RP}(\text{GapOr}) \le 1$ by outputting the bit at a uniformly random position from the input. We describe one way of seeing that $\mathsf{BPP}_{1/n}(\text{GapOr}) \ge \mathsf{WAPP}_{1/n}(\overline{\text{GapOr}}) \ge \Omega(\log n)$ provided $m \ge \log n$ (this cannot be shown via Fact 5). For $z \in \{0,1\}$, define $\mathcal{G}_z$ as the uniform distribution over $\text{GapOr}^{-1}(z)$.

▶ **Fact 9.** *For every conjunction $C\colon \{0,1\}^m \to \{0,1\}$:*
(i) *$C(\mathcal{G}_0) \in \{0,1\}$.*
(ii) *If $C(\mathcal{G}_0) = 1$ and C has width $w \le m/4$ then $C(\mathcal{G}_1) \ge 3^{-w}$.*

Proof. *(i)*: Note that $\mathcal{G}_0$ is supported entirely on the input 0^m. If C has a positive literal then $C(\mathcal{G}_0) = 0$. If C has only negative literals then $C(\mathcal{G}_0) = 1$.

(ii): Suppose C has w negative literals and no positive literals. Then

$$C(\mathcal{G}_1) = \tbinom{m-w}{m/2}/\tbinom{m}{m/2} = \tfrac{(m/2)\cdot(m/2-1)\cdots(m/2-w+1)}{m\cdot(m-1)\cdots(m-w+1)} \ge \left(\tfrac{m/2-w}{m-w}\right)^w \ge \left(\tfrac{m/2-m/4}{m-m/4}\right)^w = 3^{-w}.$$

◀

Combining Fact 7 and Fact 9 (using $h = \overline{\text{GapOr}}$, $k = m$, $\mathcal{D}_0 = \mathcal{G}_1$, $\mathcal{D}_1 = \mathcal{G}_0$, $\mathcal{D}_2$ is not needed, $\varepsilon = 1/n$, and $w = \mathsf{WAPP}_\varepsilon(h)$) implies that $3^{-w} \le \delta$, in other words $\mathsf{WAPP}_{1/n}(\overline{\text{GapOr}}) \ge \log_3(1/(2\sqrt{1/n})) \ge \Omega(\log n)$, provided $w \le m/4$. If $w > m/4$ then $\mathsf{WAPP}_{1/n}(\overline{\text{GapOr}}) \ge \Omega(\log n)$ holds anyway provided $m \ge \log n$.

Hence, our result can be restated as follows.[2]

▶ **Theorem 4 (Restated).** $\mathsf{WAPP}_\varepsilon(\text{Maj}\circ\text{GapOr}^n) \ge \Omega(n\log n)$ *for some constant $\varepsilon > 0$ provided $m \ge \log n$.*

We show $\mathsf{WAPP}_{1/36}(\text{Maj}\circ\text{GapOr}^n) > \frac{1}{16}n\log n$. By Fact 7 (using $h = \text{Maj}\circ\text{GapOr}^n$, $k = nm$, $\varepsilon = 1/36$, and $\delta = 1/3$) it suffices to exhibit distributions $\mathcal{D}_0$, $\mathcal{D}_1$, $\mathcal{D}_2$ over $h^{-1}(0)$, $h^{-1}(1)$, and $h^{-1}(0)\cup h^{-1}(1)$ respectively, such that for every conjunction C of width $\le \frac{1}{16}n\log n$, either $C(\mathcal{D}_0) > \frac{1}{3}C(\mathcal{D}_1)$ or $C(\mathcal{D}_2) > \frac{4}{3}C(\mathcal{D}_1)$ or $C(\mathcal{D}_1) = 0$. Assume n is

[2] Properties *(i)* and *(ii)* from Fact 9 are somewhat stronger than necessary for the proof of Theorem 4 to go through. The proof works, with virtually no modification, for any g satisfying the following for some distributions $\mathcal{G}_z$ over $g^{-1}(z)$ ($z \in \{0,1\}$): For every conjunction $C\colon \{0,1\}^m \to \{0,1\}$ such that $C(\mathcal{G}_0) > 0$, we have $C(\mathcal{G}_1) \le C(\mathcal{G}_0)$ and if furthermore C has width $w \le m/4$ then $C(\mathcal{G}_1) \ge 2^{-O(w)}\cdot C(\mathcal{G}_0)$.

even and for the tiebreaker, $\textsc{Maj}(y) = 1$ if $|y| = n/2$. For $\zeta \in \{0,1,2\}$ letting $\mathcal{F}_\zeta$ be the uniform distribution over all $y \in \{0,1\}^n$ with $|y| = n/2 - 1 + \zeta$ (so $\mathcal{F}_0$, $\mathcal{F}_1$, $\mathcal{F}_2$ are over $\textsc{Maj}^{-1}(0)$, $\textsc{Maj}^{-1}(1)$, $\textsc{Maj}^{-1}(1)$ respectively), define $\mathcal{D}_\zeta$ as the mixture over $y \sim \mathcal{F}_\zeta$ of $\mathcal{G}_y \coloneqq \mathcal{G}_{y_1} \times \cdots \times \mathcal{G}_{y_n}$ (i.e., $(x^1, \ldots, x^n) \sim \mathcal{G}_y$ is sampled by independently sampling $x^i \sim \mathcal{G}_{y_i}$ for all i). Put succinctly, $\mathcal{D}_\zeta \coloneqq \mathbb{E}_{y \sim \mathcal{F}_\zeta}[\mathcal{G}_y]$.

Now consider any conjunction C of width $w \le \frac{1}{16} n \log n$, and write $C(x^1, \ldots, x^n) = \prod_i C_i(x^i)$ where C_i is a conjunction. By Fact 9.*(i)*, $[n]$ can be partitioned into $A \cup B$ such that $C_i(\mathcal{G}_0) = 1$ for all $i \in A$, and $C_i(\mathcal{G}_0) = 0$ for all $i \in B$. Abbreviate $C_i(\mathcal{G}_1)$ as c_i, and for $S \subseteq [n]$ write $c_S \coloneqq \prod_{i \in S} c_i$. Identify $y \in \{0,1\}^n$ with $Y \coloneqq \{i : y_i = 1\}$, so $|y| = |Y|$. Let the uniform distribution over all size-s subsets of S be denoted by $\binom{S}{s}$, so $y \sim \mathcal{F}_\zeta$ corresponds to $Y \sim \binom{[n]}{n/2-1+\zeta}$. Let $I_{Y \supseteq B} \coloneqq \prod_{i \notin Y} C_i(\mathcal{G}_0)$ be the indicator random variable for the event $Y \supseteq B$. Now for $\zeta \in \{0,1,2\}$,

$$C(\mathcal{D}_\zeta) \;=\; \mathbb{E}_{y \sim \mathcal{F}_\zeta}[C(\mathcal{G}_y)] \;=\; \mathbb{E}_{y \sim \mathcal{F}_\zeta}\big[\textstyle\prod_i C_i(\mathcal{G}_{y_i})\big] \;=\; \mathbb{E}_{Y \sim \binom{[n]}{n/2-1+\zeta}}\big[c_Y \cdot I_{Y \supseteq B}\big]$$

$$= \underbrace{\mathbb{P}_{Y \sim \binom{[n]}{n/2-1+\zeta}}[Y \supseteq B]}_{p_\zeta} \cdot c_B \cdot \underbrace{\mathbb{E}_{S \sim \binom{A}{n/2-1+\zeta-|B|}}[c_S]}_{q_\zeta}.$$

If $c_B = 0$ then $C(\mathcal{D}_1) = 0$, so assume $c_B > 0$. Factoring out c_B and defining p_ζ and q_ζ as above (but q_ζ is undefined if $p_\zeta = 0$), our goal is to show that either $p_0 q_0 > \frac{1}{3} p_1 q_1$ or $p_2 q_2 > \frac{4}{3} p_1 q_1$ or $p_1 q_1 = 0$. There are three cases depending on whether $|B|$ is greater than, equal to, or less than $n/2$. First we collect some generally useful properties:

▷ Claim 10.

(i) $p_0 = \frac{n/2-|B|}{n/2} \cdot p_1$ and $p_1 = \frac{n/2+1-|B|}{n/2+1} \cdot p_2$.

(ii) $0 < q_1 \le \sqrt{n} \cdot q_2$ if q_1 is defined.

Proof. *(i)*: We just consider p_0 vs. p_1 since p_1 vs. p_2 is similar. Imagine sampling $Y_1 \sim \binom{[n]}{n/2}$ and then obtaining the set Y_0 by removing a uniformly random $i \in Y_1$. If $Y_1 \supseteq B$, then $Y_0 \supseteq B$ when $i \in Y_1 \setminus B$, which happens with probability $\frac{n/2-|B|}{n/2}$ (assuming $|B| \le n/2$; if $|B| > n/2$ then $p_0 = p_1 = 0$). Thus

$$p_0 \;=\; \mathbb{P}[Y_0 \supseteq B] \;=\; \mathbb{P}[Y_0 \supseteq B \,|\, Y_1 \supseteq B] \cdot \mathbb{P}[Y_1 \supseteq B] \;=\; \tfrac{n/2-|B|}{n/2} \cdot p_1.$$

(ii): Let w_i be the width of C_i, so $\sum_i w_i = w \le \frac{1}{16} n \log n$. Then $w_i \le \frac{1}{4} \log n \le m/4$ for at least $3n/4$ many values of i, and for such i note that by Fact 9.*(ii)*, $c_i \ge 3^{-(\log n)/4} \ge n^{-2/5}$ if $i \in A$. This implies that if we sample a uniformly random i from any $A' \subseteq A$ with $|A'| = n/2$ (note that $|A| \ge n/2$ if q_1 is defined) then $\mathbb{E}_{i \in A'}[c_i] \ge \frac{1}{2} \cdot n^{-2/5} + \frac{1}{2} \cdot 0 \ge 1/\sqrt{n}$. Now to relate q_2 and q_1,

$$q_2 \;=\; \mathbb{E}_{S \sim \binom{A}{n/2-|B|}}\big[c_S \cdot \mathbb{E}_{i \in A \setminus S}[c_i]\big] \;\ge\; \mathbb{E}_{S \sim \binom{A}{n/2-|B|}}\big[c_S/\sqrt{n}\big] \;=\; q_1/\sqrt{n}$$

where the inequality uses $|A \setminus S| = (n - |B|) - (n/2 - |B|) = n/2$. Furthermore, $q_1 > 0$ if q_1 is defined, because $n/2 - |B| \le |A| - n/4$ and thus there exists an $S \subseteq A$ with $|S| = n/2 - |B|$ and $c_i \ge n^{-2/5} > 0$ for all $i \in S$, hence $c_S > 0$. (A similar argument shows $0 < q_0 \le \sqrt{n} \cdot q_1$ if q_0 is defined, but we will not need that.) ◁

Case $|B| > n/2$. In this case, $p_1 = 0$ so we are done.

Case $|B| = n/2$. By Claim 10, $p_2 = p_1 \cdot (n/2 + 1)$ and $q_2 \ge q_1/\sqrt{n} > 0$ and thus

$$p_2 q_2 \;\ge\; p_1 q_1 \cdot (n/2+1)/\sqrt{n} \;>\; \tfrac{4}{3} p_1 q_1.$$

Case $|B| < n/2$. We will show that $\frac{p_0}{p_1} \geq \frac{1}{2} \cdot \frac{p_1}{p_2}$ and $\frac{q_2}{q_1} \geq \frac{9}{10} \cdot \frac{q_1}{q_0}$, which yields the punchline:

If $p_0 q_0 \leq \frac{1}{3} p_1 q_1$ then $\frac{q_2}{q_1} \geq \frac{9}{10} \cdot \frac{q_1}{q_0} \geq \frac{9}{10} \cdot 3 \cdot \frac{p_0}{p_1} \geq \frac{9}{10} \cdot 3 \cdot \frac{1}{2} \cdot \frac{p_1}{p_2} > \frac{4}{3} \cdot \frac{p_1}{p_2}$ and thus $p_2 q_2 > \frac{4}{3} p_1 q_1$.

First, $\frac{p_0}{p_1} \geq \frac{1}{2} \cdot \frac{p_1}{p_2}$ follows from Claim 10.*(i)* using $|B| \leq n/2 - 1$:

$$\frac{p_0}{p_1} \;=\; \frac{n/2+1}{n/2} \cdot \frac{n/2-|B|}{n/2+1-|B|} \cdot \frac{p_1}{p_2} \;\geq\; 1 \cdot \frac{n/2-(n/2-1)}{n/2+1-(n/2-1)} \cdot \frac{p_1}{p_2} \;=\; \frac{1}{2} \cdot \frac{p_1}{p_2}.$$

It just remains to show $\frac{q_2}{q_1} \geq \frac{9}{10} \cdot \frac{q_1}{q_0}$. Henceforth let $s \coloneqq n/2 - 1 - |B| \geq 0$. The experiment $S \sim \binom{A}{s+2}$ in the definition of q_2 can alternatively be viewed as:

- Sample $S_0 \sim \binom{A}{s}$.
- Sample $i \in A \setminus S_0$ u.a.r. and let $S_1 \coloneqq S_0 \cup \{i\}$.
- Sample $j \in A \setminus S_1$ u.a.r. and let $S = S_2 \coloneqq S_1 \cup \{j\}$.

That is, i and j are sampled without replacement. We consider an "ideal" (easier to analyze) version of this experiment that samples i and j with replacement, in other words, the third step becomes:

- Sample $j \in A \setminus S_0$ u.a.r. and let $S_2^* \coloneqq S_1 \cup \{j\}$.

Now S_2^* is a *multiset*, which may have two copies of i, in which case the product $c_{S_2^*}$ has two factors of c_i. Just as $q_2 \coloneqq \mathbb{E}[c_{S_2}]$, we let $q_2^* \coloneqq \mathbb{E}[c_{S_2^*}]$, and we next show how to derive $\frac{q_2^*}{q_1} \geq \frac{q_1}{q_0}$ from the following claim:

▷ Claim 11. For all nonnegative numbers $\alpha_1, \ldots, \alpha_N$ and $\beta_1, \ldots, \beta_N$ such that $\alpha_k \beta_k > 0$ for some k,

$$\frac{\sum_k \alpha_k \beta_k^2}{\sum_k \alpha_k \beta_k} \;\geq\; \frac{\sum_k \alpha_k \beta_k}{\sum_k \alpha_k}.$$

Proof. By clearing denominators, this inequality is equivalent to

$$\big(\textstyle\sum_k \alpha_k\big)\big(\sum_k \alpha_k \beta_k^2\big) \;\geq\; \big(\sum_k \alpha_k \beta_k\big)^2$$

which can be rewritten as

$$\textstyle\sum_{k,\ell} \alpha_k \alpha_\ell \beta_\ell^2 \;\geq\; \sum_{k,\ell} \alpha_k \beta_k \alpha_\ell \beta_\ell.$$

Subtracting $\sum_k \alpha_k^2 \beta_k^2$ from both sides, this is equivalent to

$$\textstyle\sum_{k<\ell} \big(\alpha_k \alpha_\ell \beta_\ell^2 + \alpha_\ell \alpha_k \beta_k^2\big) \;\geq\; \sum_{k<\ell} 2 \alpha_k \beta_k \alpha_\ell \beta_\ell.$$

We show that this inequality holds for each summand separately. Factoring out $\alpha_k \alpha_\ell$, this reduces to showing $\beta_\ell^2 + \beta_k^2 \geq 2 \beta_k \beta_\ell$, which holds since

$$\beta_\ell^2 + \beta_k^2 - 2 \beta_k \beta_\ell \;=\; (\beta_\ell - \beta_k)^2 \;\geq\; 0. \qquad \triangleleft$$

In the statement of Claim 11, let the index k correspond to S_0, let $N \coloneqq \binom{|A|}{s}$, let $\alpha_k \coloneqq c_{S_0}/N$, and let $\beta_k \coloneqq \mathbb{E}_{i \in A \setminus S_0}[c_i]$. Then

$$q_0 = \textstyle\sum_k \alpha_k \quad \text{and} \quad q_1 = \sum_k \alpha_k \beta_k \quad \text{and} \quad q_2^* = \sum_k \alpha_k \beta_k^2$$

and $q_0 \geq q_1 > 0$ by Claim 10.*(ii)* (i.e., $\alpha_k \beta_k > 0$ for some k) so by Claim 11 we indeed have $\frac{q_2^*}{q_1} \geq \frac{q_1}{q_0}$. To conclude that $\frac{q_2}{q_1} \geq \frac{9}{10} \cdot \frac{q_1}{q_0}$, we just need to show $q_2 \geq \frac{9}{10} q_2^*$.

The third step of the S_2 experiment is just the third step of the S_2^* experiment conditioned on $j \neq i$, which happens with probability $1 - \frac{1}{|A|-s}$. With probability $\frac{1}{|A|-s}$, we get $j = i$ in

the S_2^* experiment. If we condition on the latter event, it yields another experiment, whose result we call S_2^{err}, which is a multiset definitely containing two copies of i. Correspondingly we define $q_2^{\mathrm{err}} \coloneqq \mathbb{E}[c_{S_2^{\mathrm{err}}}]$ (with two factors of c_i). Now we have

$$q_2^* \;=\; \mathbb{P}[j \neq i]\cdot\mathbb{E}\big[c_{S_2^*} \,\big|\, j \neq i\big] + \mathbb{P}[j = i]\cdot\mathbb{E}\big[c_{S_2^*} \,\big|\, j = i\big] \;=\; \big(1 - \tfrac{1}{|A|-s}\big)\cdot q_2 + \tfrac{1}{|A|-s}\cdot q_2^{\mathrm{err}} \;\leq\; q_2 + \tfrac{2}{n}\cdot q_2^{\mathrm{err}}$$

since $|A| - s = (n - |B|) - (n/2 - 1 - |B|) = n/2 + 1 \geq n/2$.

The S_2^{err} experiment can alternatively be viewed as:

- Sample $S_1 \sim \binom{A}{s+1}$.
- Sample $i \in S_1$ u.a.r. and let $S_2^{\mathrm{err}} \coloneqq S_1 \cup \{i\}$.

This implies that $q_2^{\mathrm{err}} \leq q_1$ because the extra factor of $c_i \leq 1$ cannot increase the expectation. By Claim 10.*(ii)* we get $q_2^{\mathrm{err}} \leq q_1 \leq \sqrt{n}\cdot q_2$. Combining, we have

$$q_2^* \;\leq\; q_2 + \tfrac{2}{n}\cdot\sqrt{n}\cdot q_2 \;=\; \big(1 + \tfrac{2}{\sqrt{n}}\big)q_2 \;\leq\; \tfrac{10}{9}q_2$$

and thus $q_2 \geq \frac{9}{10}q_2^*$ as desired. This concludes the proof of Theorem 4.

4 Open questions

▶ **Open Question 12.** *Is there a total function* $g\colon \{0,1\}^m \to \{0,1\}$ *such that* $\mathsf{BPP}(\textsc{Xor}\circ g^n) \geq \Omega(n\log n\cdot\mathsf{BPP}(g))$ *or* $\mathsf{BPP}(\textsc{Maj}\circ g^n) \geq \Omega(n\log n\cdot\mathsf{BPP}(g))$*?*

Since Fact 9 captures the only properties of $g = \textsc{GapOr}$ used in our proof of Theorem 4, this provides a possible roadmap for confirming Open Question 12: Just find a total function g satisfying properties similar to Fact 9, enabling our proof of Theorem 4 to go through. However, such a g would need to have certificate complexity $\omega(\mathsf{BPP}(g))$, and it remains a significant open problem to find any such total function g (the "pointer function" [19, 2] and "cheat sheet" [1] methods do not seem to work).

Another approach for confirming Open Question 12 would be to generalize the strong direct sum theorem from [7] to show that $\mathsf{BPP}(\textsc{Xor}\circ g^n) \geq \Omega(n\cdot\overline{\mathsf{BPP}}_{1/n}(g))$ or $\mathsf{BPP}(\textsc{Maj}\circ g^n) \geq \Omega(n\cdot\overline{\mathsf{BPP}}_{1/n}(g))$ holds for all g. This would answer Open Question 12 in the affirmative, since [7] designed a total function g satisfying $\overline{\mathsf{BPP}}_{1/n}(g) \geq \Omega(\mathsf{RP}(g)\cdot\log n)$ using the "pointer function" method. Compared to our approach from the previous paragraph, this approach involves less stringent requirements on g, which makes it easier to design g but harder to prove the composition lower bound.

▶ **Open Question 13.** *Is there a total function* $f\colon \{0,1\}^n \to \{0,1\}$ *such that* $\mathsf{BPP}^*(f) \geq \omega(\mathsf{BPP}^\dagger(f))$ *(or similarly,* $\mathsf{BPP}(f\circ\textsc{GapMaj}^n) \geq \omega(\mathsf{BPP}(f\circ\textsc{GapOr}^n))$*)?*

It is not difficult to find such a *partial* function f. Namely, take any function $f'\colon \{0,1\}^n \to \{0,1\}$ such that $\mathsf{BPP}^*(f') \geq \Omega(n\log n)$, such as $f' = \textsc{Xor}$ or $f' = \textsc{Maj}$. Then take $f = f'\circ\textsc{Which}^n$, which has input length $2n$ (recall from Subsection 1.1 that given $y \in \{0,1\}^2$ with the promise that y has Hamming weight 1, $\textsc{Which}(y)$ indicates the location of the unique 1 in y). A simple reduction shows $\mathsf{BPP}^*(f) \geq \mathsf{BPP}^*(f')$. However, $\mathsf{BPP}^\dagger(f) \leq O(n)$: For each block of 2 bits, we can repeatedly query both until one of them returns 1 (which takes $O(1)$ queries in expectation). After doing this for all n blocks (which takes $O(n)$ queries in expectation), we know for sure what the entire actual input is. By Markov's inequality, we can abort the execution after $O(n)$ queries while introducing only a small constant error probability. (Intuitively, composition with $\textsc{Which}$ preserves hardness for 2-sided noise but converts 1-sided noise to "0-sided noise", and no partial function needs $\omega(n)$ queries in the setting of 0-sided noise.)

In communication (rather than query) complexity, somewhat analogous questions have been studied in specific contexts [24, 8, 26]. The proof of Theorem 2 also works for communication complexity. It would be interesting to develop analogues of Theorem 3 and Theorem 4 for communication complexity.

References

1 Scott Aaronson, Shalev Ben-David, and Robin Kothari. Separations in query complexity using cheat sheets. In *Proceedings of the 48th Symposium on Theory of Computing (STOC)*, pages 863–876. ACM, 2016. doi:10.1145/2897518.2897644.

2 Andris Ambainis, Kaspars Balodis, Aleksandrs Belovs, Troy Lee, Miklos Santha, and Juris Smotrovs. Separations in query complexity based on pointer functions. *Journal of the ACM*, 64(5):32:1–32:24, 2017. doi:10.1145/3106234.

3 Anurag Anshu, Dmitry Gavinsky, Rahul Jain, Srijita Kundu, Troy Lee, Priyanka Mukhopadhyay, Miklos Santha, and Swagato Sanyal. A composition theorem for randomized query complexity. In *Proceedings of the 37th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, pages 10:1–10:13. Schloss Dagstuhl, 2017. doi:10.4230/LIPIcs.FSTTCS.2017.10.

4 Andrew Bassilakis, Andrew Drucker, Mika Göös, Lunjia Hu, Weiyun Ma, and Li-Yang Tan. The power of many samples in query complexity. In *Proceedings of the 47th International Colloquium on Automata, Languages, and Programming (ICALP)*. Schloss Dagstuhl, 2020. To appear.

5 Shalev Ben-David and Eric Blais. A tight composition theorem for the randomized query complexity of partial functions. Technical Report 2002.10809, arXiv, 2020. arXiv:2002.10809.

6 Shalev Ben-David and Robin Kothari. Randomized query complexity of sabotaged and composed functions. *Theory of Computing*, 14(1):1–27, 2018. doi:10.4086/toc.2018.v014a005.

7 Eric Blais and Joshua Brody. Optimal separation and strong direct sum for randomized query complexity. In *Proceedings of the 34th Computational Complexity Conference (CCC)*, pages 29:1–29:17. Schloss Dagstuhl, 2019. doi:10.4230/LIPIcs.CCC.2019.29.

8 Eric Blais, Joshua Brody, and Badih Ghazi. The information complexity of hamming distance. In *Proceedings of the 18th International Workshop on Randomization and Computation (RANDOM)*, pages 465–489. Schloss Dagstuhl, 2014. doi:10.4230/LIPIcs.APPROX-RANDOM.2014.465.

9 Harry Buhrman and Ronald de Wolf. Complexity measures and decision tree complexity: A survey. *Theoretical Computer Science*, 288(1):21–43, 2002. doi:10.1016/S0304-3975(01)00144-X.

10 Chris Cade. Post-selected classical query complexity. Technical report, arXiv, 2018. arXiv:1804.10010.

11 Chinmoy Dutta and Jaikumar Radhakrishnan. Lower bounds for noisy wireless networks using sampling algorithms. In *Proceedings of the 49th Symposium on Foundations of Computer Science (FOCS)*, pages 394–402. IEEE, 2008. doi:10.1109/FOCS.2008.72.

12 William Evans and Nicholas Pippenger. Average-case lower bounds for noisy Boolean decision trees. *SIAM*, 28(2):433–446, 1998. doi:10.1137/S0097539796310102.

13 Uriel Feige, Prabhakar Raghavan, David Peleg, and Eli Upfal. Computing with noisy information. *SIAM Journal on Computing*, 23(5):1001–1018, 1994. doi:10.1137/S0097539791195877.

14 Dmitry Gavinsky, Troy Lee, Miklos Santha, and Swagato Sanyal. A composition theorem for randomized query complexity via max-conflict complexity. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 64:1–64:13. Schloss Dagstuhl, 2019. doi:10.4230/LIPIcs.ICALP.2019.64.

15 Dmitry Gavinsky and Shachar Lovett. En route to the log-rank conjecture: New reductions and equivalent formulations. In *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 514–524. Springer, 2014. doi:10.1007/978-3-662-43948-7_43.

16 Mika Göös and T. S. Jayram. A composition theorem for conical juntas. In *Proceedings of the 31st Computational Complexity Conference (CCC)*, pages 5:1–5:16. Schloss Dagstuhl, 2016. doi:10.4230/LIPIcs.CCC.2016.5.

17 Mika Göös, T. S. Jayram, Toniann Pitassi, and Thomas Watson. Randomized communication vs. partition number. *ACM Transactions on Computation Theory*, 10(1):4:1–4:20, 2018. doi:10.1145/3170711.

18 Mika Göös, Shachar Lovett, Raghu Meka, Thomas Watson, and David Zuckerman. Rectangles are nonnegative juntas. *SIAM Journal on Computing*, 45(5):1835–1869, 2016. doi:10.1137/15M103145X.

19 Mika Göös, Toniann Pitassi, and Thomas Watson. Deterministic communication vs. partition number. *SIAM Journal on Computing*, 47(6):2435–2450, 2018. doi:10.1137/16M1059369.

20 Navin Goyal and Michael Saks. Rounds vs. queries tradeoff in noisy computation. *Theory of Computing*, 6(1):113–134, 2010. doi:10.4086/toc.2010.v006a006.

21 Rahul Jain, Hartmut Klauck, and Miklos Santha. Optimal direct sum results for deterministic and randomized decision tree complexity. *Information Processing Letters*, 110(20):893–897, 2010. doi:10.1016/j.ipl.2010.07.020.

22 Jedrzej Kaniewski, Troy Lee, and Ronald de Wolf. Query complexity in expectation. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 761–772. Springer, 2015. doi:10.1007/978-3-662-47672-7_62.

23 Claire Kenyon and Valerie King. On Boolean decision trees with faulty nodes. *Random Structures and Algorithms*, 5(3):453–464, 1994. doi:10.1002/rsa.3240050306.

24 Marco Molinaro, David Woodruff, and Grigory Yaroslavtsev. Beating the direct sum theorem in communication complexity with implications for sketching. In *Proceedings of the 24th Symposium on Discrete Algorithms*, pages 1738–1756. ACM-SIAM, 2013. doi:10.1137/1.9781611973105.125.

25 Ilan Newman. Computing in fault tolerant broadcast networks and noisy decision trees. *Random Structures and Algorithms*, 34(4):478–501, 2009. doi:10.1002/rsa.20240.

26 Mert Saglam. Near log-convexity of measured heat in (discrete) time and consequences. In *Proceedings of the 59th Symposium on Foundations of Computer Science (FOCS)*, pages 967–978. IEEE, 2018. doi:10.1109/FOCS.2018.00095.

27 Alexander Sherstov. Making polynomials robust to noise. *Theory of Computing*, 9:593–615, 2013. doi:10.4086/toc.2013.v009a018.

A Proof of Theorem 2: Or never necessitates amplification

For completeness, we provide a self-contained proof that $\mathsf{BPP}^*(\textsc{Or}) \leq O(n)$, using the following standard fact about random walks ("the drunkard at the cliff").

▶ **Lemma 14.** *Consider a random walk on the integers that begins at* 0 *and in each step moves right* $(+1)$ *with probability* p *and moves left* (-1) *with probability* $1-p$.

(i) *If* $p < 1/2$ *then the expected time at which the walk first visits* -1 *is* $1/(1-2p)$.

(ii) *If* $p > 1/2$ *then the probability that the walk ever visits* -1 *is* $(1-p)/p$.

Proof of Lemma 14. *(i)*: If random variable X represents the time at which the walk first visits -1, then its expectation satisfies $\mathbb{E}[X] = 1 + p \cdot 2\mathbb{E}[X]$ since after the first step, it either is already at -1, or is at $+1$ in which case to reach -1 it must first get back to 0 ($\mathbb{E}[X]$ expected time) then from there get to -1 (another $\mathbb{E}[X]$ expected time). This equation has a unique solution $\mathbb{E}[X] = 1/(1-2p) < \infty$.

(ii): If event E represents the walk ever visiting -1, then its probability satisfies $\mathbb{P}[E] = (1-p) \cdot 1 + p \cdot \mathbb{P}[E]^2$ since after the first step, it either is already at -1, or is at $+1$ in which case to reach -1 it must first get back to 0 (probability $\mathbb{P}[E]$) then from there get to -1 (again probability $\mathbb{P}[E]$). This equation has two solutions $\mathbb{P}[E] \in \{(1-p)/p, 1\}$. To rule out

$\mathbb{P}[E] = 1$, we define q_k as the probability that the walk visits -1 within the first k steps, and we show by induction on k that $q_k \leq (1-p)/p$. The base case is trivial since $q_0 = 0$. Assuming $q_k \leq (1-p)/p$ we show $q_{k+1} \leq (1-p)/p$. After the first step, with probability $1-p$ it is already at -1, and with probability p it is at $+1$. In the latter case, to get to -1 within a total of $k+1$ steps (including the first step), it must get from $+1$ to 0 and then from there it must get to -1, all within k more steps; in particular, the walk must get from $+1$ to 0 within k steps (probability $\leq q_k$) and then from 0 to -1 within k steps (probability $\leq q_k$). Overall we can bound $q_{k+1} \leq (1-p) \cdot 1 + p \cdot q_k^2 \leq (1-p) + p \cdot (1-p)^2/p^2 = (1-p)/p$. ◀

Proof of Theorem 2. We may assume the noise probabilities are $\leq 1/4$ (rather than just $\leq 1/3$), because whenever an input bit is queried, we can instead query it five times and pretend that the majority vote was the result of the single query. This would only affect the cost by a constant factor. With this assumption, here is our decision tree, on input $y \in \{0,1\}^n$:

For $i = 1, 2, \ldots, n$:
  Repeat:
    Query y_i.
    If the queries to y_i have resulted in more 0s than 1s so far,
      then break out of the inner loop.
    If a total of $6n$ queries have been made (across all input bits), then halt and output 1.
Halt and output 0.

This decision tree's cost is $\leq 6n$. To see the correctness, consider any input $y \in \{0,1\}^n$ and any tuple of noise probabilities $(\nu_1, \ldots, \nu_n)$ where each $\nu_i \leq 1/4$. For each i, the random variable

"number of 1s minus number of 0s, among the queries to y_i so far"

is a random walk with move-right probability $p_i = \nu_i \leq 1/4$ if $y_i = 0$ and $p_i = 1 - \nu_i \geq 3/4$ if $y_i = 1$, and which stops when it visits -1.

First assume $\textsc{Or}(y) = 0$. Then for each i, $y_i = 0$ and so by Lemma 14.*(i)*, the expected number of queries until the inner loop is broken is $1/(1-2p_i) \leq 2$. By linearity, the expected total number of queries until all n inner loops have been broken is $\leq 2n$, so by Markov's inequality this number of queries is $< 6n$ with probability $\geq 2/3$. Thus the decision tree outputs 0 with probability $\geq 2/3$.

Now assume $\textsc{Or}(y) = 1$. Then for some i, $y_i = 1$ and so by Lemma 14.*(ii)*, with probability $1 - (1-p_i)/p_i = 2 - 1/p_i \geq 2/3$ there would never be more 0s than 1s from the queries to y_i. In that case, the decision tree would never break out of the i^{th} inner loop, even if it were allowed to run forever. Thus the decision tree outputs 1 with probability $\geq 2/3$. ◀